Geometric detail suppression by the Fourier transform

Yong-Gu Lee†* and Kunwoo Lee‡

As the Finite Element Method is widely used in strength analysis, automatic mesh generation draws more attention these days. For a given geometric tolerance value, the purpose of mesh generators is to discretize the continuous model within this error limit. Faithfulness to this condition produces many small elements at small features. Often, these regions are of little interest and computer resources are thus wasted. It is wished to suppress selectively, small features from the model before discretizing. This can be achieved by low-pass filtering a CAD model. A method to apply the techniques in signal processing to the manipulation of a three-dimensional model is proposed. © 1998 Elsevier Science Ltd. All rights reserved

Keywords: detail removal, detail suppression, finite element meshing, Fourier transform, signal processing

INTRODUCTION

Humans have many intelligent characteristics. One of these is the ability to view a complex geometric model in different levels of details. At the first glimpse they see the overall shape and then eventually look into the various detail features. This intelligence is not easy to theorize. The key element that distinguishes the large features from the small features is the size, i.e. the (local) area in two dimensions (2D) or the (local) volume in three dimensions (3D). It is desirable to analyze a geometric model through this metric size. Then we can find out which part can be considered suppressible in relation to the overall shape. This can be achieved by one of the signal processing method, the Fourier transform.

Electrical signals often carry noises that are caused by environmental disturbances. Noises are small abrupt changes in the amplitude of signal at a short duration. Noises are equivalent to the local area or volume we previously explained. It is well known that any function (any signal) can be represented by a combination of many sinusoidal functions of different frequencies. Using this fact, noises are smoothed out by discarding the high frequency sinusoidal functions of different frequencies. Using this fact, noises are smoothed out by discarding the high frequency sinusoidal functions of different frequencies. This is solved by introducing the Fourier transform

Fourier transform is used to decompose a function into the weighted sum of frequency varying sinusoidal functions. If we limit the classes of shapes that we deal with to shapes that can be represented by a function \( y = f(x) \), we can eliminate small features by the previously explained method. However, a shape illustrated in Figure 1 cannot be represented by this scheme.

The difficult part is in representing a solid model with arbitrary shape by a function that can be decomposed into a combination of sinusoidal functions. This is solved by introducing a spatially defined value function and regarding the region where the values are larger than certain value as the inside of a given solid model. It is easy to understand this concept by thinking the value as a temperature. We viewed a solid model as a life form. A typical life form has a higher temperature than its surroundings as shown in Figure 2. Modeling the temperature distribution of a life form and its surroundings by sinusoidal functions is very natural.

REVIEW

Detail removal in FEM

There have been many discussions about the need of the detail removal in the finite element generation literature, but few solutions were given. Some predicted the potential methods for detail removal \(^1–4\), others proposed a feature recognition approach \(^5\) and a method based on quantifying the detail features \(^6, 7\). While these works are for solid meshing, there is also a work for removing small features on a surface model for shell meshing \(^8\).

Shephard \(^1\) suggested to mark detail features and ignore them during the mesh generation stage which could be implemented with little change in the mesh generation program. Shephard et al. \(^2, 3\) suggested that the mesh generation tools provide geometric modeling operators so users can suppress unimportant features. He also suggested a method to suppress small holes in the following steps; Delete all the holes to get a simply connected shape and derive the stress distribution without the holes. Then estimate the stresses around the inner holes by multiplying the stress value by the stress concentration factor. If the value is too high, re-mesh the shape including the holes having high stress values and analyze again.

Finnigan et al. \(^4\) noted a computational gain in suppressing detail features prior to the automatic mesh generation. He mentioned some heuristics will be required to achieve this. He stated as a starting point, the analysis system can provide feature-elimination tools. This will be better than
Geometric detail suppression by the Fourier transform: Y G Lee, K Lee

Figure 1 Shape that cannot be represented by function

having nothing, but a geometric reasoning method will be essential to achieve complete automation.

Dabke et al. 7 built an expert system for removing detail features. Their method relies on the feature recognition techniques and they admit the limitation of these technique. Armstrong et al. 6,7 introduced a metric to distinguish small features from the larger ones. They defined dimensionless ratios relating the size of holes, edges and more complex features such as notches and protrusions to the radius of the neighboring inscribed disc. Their method establishes a distinction form the others in that it is a metric system for detail features. In other words, two features can be compared uniquely to decide which one is more on the detail side than the other. This ranks features unambiguously. We believe a feature can not be concluded to a detail one or not without comparing to other features. Our method resembles this method in that it is also based on a metric system. We use Fourier transform to measure the detailness.

Rezayat 8, in his mid-surface extraction algorithm, describes removing small holes on a mid-surface based on the diameters and removing mid-surfaces based on the form of underlying surface or size. He notes that great improvements in meshing can be expected by doing so.

Fourier transform in geometric applications

There have been several Fourier transform applications in the geometric domain. We will briefly describe them below. Oh and Yoon 9 applied the Fourier transform in the die (blocker) design. In forging operations a simple shape is transformed to the final complex shape by dies. These intermediate dies are called blockers, which have the negative shapes of several simplified stages of the final shape. At the end, the final shape is obtained by the finishing blocker, which has the negative shape of the part to be made. Using this blocker, the excessive deformation which is the main reason for the part defect can be avoided. Oh and Yoon proposed to derive the blocker shape by expanding the finisher geometry in terms of Fourier series, eliminating the higher frequency terms, and inverse transforming the Fourier series. In this process, sharp corners in the dies are also smoothed so that proper metal distribution and flow are ensured. This method is only applicable to the shapes whose boundary can be represented by a non-parametric explicit function. Park and Lee 10 proposed a 3D Fourier descriptor (FD3) for shape representation. The FD3 is a double Fourier transform of serial cross-sectional contours of a shape in 3D. Cylindrical coordinate system (r,θ, z) is used for a 3D Fourier descriptor and polar coordinate (r,θ) is used for a 2D Fourier descriptor (FD2). For example, r = 2 + sin(θ) is a simple FD2 representing a flower like loop. At each cross-section 2D Fourier descriptors are calculated and these are Fourier transformed again with respect to the height. Only single shell can be represented by this method. Roach and Martin 11 proposed to represent the boundary of a 2D shape as parametric functions of x(s) and y(s), where s is an arc length or some other suitable parameter. Each of these functions would then be low-pass filtered. This method requires two one-dimensional Fourier transforms. As a result, the boundary is smoothed. Note that this representation can handle arbitrary boundary shapes. Extension of this method to 3D involves the consideration of the topology of the objects being smoothed. In 2D, parameterization follows simply along the edge sequence of the boundary. In 3D, however, the parameterization does not run naturally from one face to another. They provide some ideas about the possible parameterizations.

Taubin 12 used low-pass filtering to smooth polyhedral surfaces. In his method, each vertex location is changed to the weighted average of neighboring vertex locations. Averaging has the effect of low-pass filtering.

Detail removal in computer graphics

Surprisingly, detail removal by low-pass filtering is an important issue in the computer graphics (CG). Detail removal in CG is realized as a by-product of antialiasing operation. Antialiasing 13 is a method to hide jagged bitmap image from the human eye. Technically, it is changing a pixel value to the weighted average of its neighbors. It blurs the image consequently. Theoretically, it is applying the low-pass filter to the image. Low-pass filtering in the frequency domain correlates to convolution of a low-pass filter in the spatial domain 14. The convolution kernel used here is the inverse transform of the low-pass filter in the frequency domain. Many researchers prefer to use the convolution method as it requires smaller memory. These concepts can be extended to 3D. Three-dimensional antialiasing is a new terminology used in one of the recent CG frontiers called volume graphics 15. We briefly explain what this new sub-field of CG is to explain the detail removal researches in this field.

Figure 2 Conversion of shape into temperature distribution function

Figure 3 Three types of models
The major concerns of volume graphics are the storing, visualizing, and surface, extracting of volumetric data sets. These data sets have values associated to the grid points in the Cartesian coordinate system. These values can represent many other properties. But for our discussion, readers can simply think of them as densities. Then we can think the aggregate of points whose values are greater than certain prescribed value as a solid model. These data sets when naively drawn as a point on the computer screen would look very flat and lose the three dimensional characteristic inherent in the data. To achieve better result without giving much efforts, a very small box can be drawn for each point. This method gives less flat result, but the scene would look very jagggy. The primary concern of antialiasing in 3D voxelized graphics is to produce a smooth scene while preserving the feeling of depth. In other words, it is desired to obtain an alias-free 3D models. In implementation, each grid data value is replaced by an average value of its neighbors. This is the 3D antialiasing. After the antialiasing, a polyhedral surface can be fitted on the iso-surface (this is a surface constructed by grouping the points having the same value) of the data sets by a triangle mesh fitting technique, such as the marching cubes. Taosong et al. suggested a method for a simplification of volumetric objects. It is accomplished by sampling and low-pass filtering the object into multi-resolution volume buffers and applying the marching cubes algorithm to generate a multi-resolution triangle-mesh hierarchy. This produces a polygonal surface which is much improved from the jaggy boxes. This method is regarded as a detail removal method since it removes small features in the model. Unfortunately, this method cannot output infinitely high frequencies, such as those introduced by sharp edges and vertices.

The similarity between our approach and the approach in computer graphics lies in the intermediate representation, the filtered object domain. CG methods apply low-pass filtering the object domain by convolution whereas our method does in the frequency domain to achieve this.

### Table 1: Definition of \( g(x,y,z) \)

<table>
<thead>
<tr>
<th>( x,y,z ) conditions</th>
<th>( g(x,y,z) = )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a = -1 \left( \frac{a}{b} \leq x \leq LT \right) ) or ( 0 \leq x &lt; \frac{a}{b} )</td>
<td>( \frac{\text{oga} - 1}{2} \left( (a - LT) + \frac{\text{oga} + 1}{2} \right) )</td>
</tr>
<tr>
<td>( b = -1 \left( \frac{b}{c} \leq y \leq MT \right) ) or ( 0 \leq y &lt; \frac{b}{c} )</td>
<td>( \frac{\text{hbb} - 1}{2} \left( (b - MT) + \frac{\text{hbb} + 1}{2} \right) )</td>
</tr>
<tr>
<td>( c = -1 \left( \frac{c}{d} \leq z \leq NT \right) ) or ( 0 \leq z &lt; \frac{c}{d} )</td>
<td>( \frac{\text{cc} - 1}{2} \left( (c - NT) + \frac{\text{cc} + 1}{2} \right) )</td>
</tr>
</tbody>
</table>

**Figure 4** Detail face selection scheme (a) (b) (c)

**Figure 5** Input geometry
Geometric detail suppression by the Fourier transform: Y G Lee, K Lee

Figure 6  Object domain (a) (b)

Although different methods are used, two approaches get the same result. The difference is in the final output representations. CG methods get facets whose vertices' values are a certain prescribed value in the filtered object domain. These facets in aggregate are called an iso-surface. However, we use the filtered object domain as a mean to rank the boundary elements from a major shape element to a detail shape element. Subsequently, we delete those classified as detail and repair the wounds (caused by the deletion) to obtain a complete solid model. Since we directly deal with the boundary elements of CAD model, we preserve sharp edges and corners as opposed to Taosong et al.’s method. For the remaining paragraphs, we describe some literature not directly related to the detail removal, but that requires some attention.

There is an interesting morphing technique by Hughes 19. It is based on interpolating smoothly between the Fourier transforms of the two volumetric models and then transforming the results back. The high frequency of the first model are gradually removed, the low frequencies are interpolated to those of the second, and the high frequencies of the second model are gradually added in.

There are other signal processing methods. Readers should note wavelets. Wavelets 20 are mathematical tool for hierarchical decomposing functions. They allow a function to be described in terms of a coarse overall shape, plus details that range from broad to narrow. Wavelets are better suited for computer calculations and its theoretical basis comes from digital signal processing. Muraki 21 applied wavelet transform to volume data generated from a set of magnetic resonance images. His method can be used to compress a volumetric data sets.

**METHOD OF APPROACH**

A method to represent an arbitrary 3D geometric model by the Fourier basis functions is given. A geometric reasoning algorithm to detect detail geometric boundary element using this volumetric representation is then given.

**Low pass filtering a geometric model**

It is well known that a function \( h(x,y,z) \), which is defined in the geometric domain, can be uniquely represented in the frequency domain.

Figure 7  Frequency domain

680
frequency domain by the Fourier transform, $H(u, v, w)$,

$$H(u, v, w) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x, y, z) e^{-i2\pi(ux + vy + wz)} \, dx \, dy \, dz$$

(1)

where $i = \sqrt{-1}$. The inverse transform is defined by

$$h(x, y, z) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} H(u, v, w) e^{i2\pi(ux + vy + wz)} \, du \, dv \, dw.$$ 

(2)

By representing a 3D geometric model by $h(x, y, z)$ in eqn (3) and low-pass filtering, a simplified geometric model can be obtained. $T_0$ is a constant value which divides the geometric model and the surrounding space.

$$h(x, y, z) \geq T_0, \text{ when } (x, y, z) \text{ is inside the volume}$$

(3)

$$h(x, y, z) < T_0, \text{ when } (x, y, z) \text{ is outside the volume}.$$ 

The rest of this section discusses how to use the analytic equations (1-3) in a computer program. The readers can jump to the next section without losing anything.

By selecting $T_L, T_L$ and $T_L$ so that they contain the volume, the volume is thought as one period of a periodic volume function whose periods are $T_L, T_L$ and $T_L$. The Fourier transform given in eqn (1) can then be represented by eqn (4). This means the Fourier transform is applied to this one period of the volume function.

$$H(u, v, w) = \int_{-\frac{T_L}{2}}^{\frac{T_L}{2}} \int_{-\frac{T_L}{2}}^{\frac{T_L}{2}} \int_{-\frac{T_L}{2}}^{\frac{T_L}{2}} h(x, y, z) e^{-i2\pi(ux + vy + wz)} \, dx \, dy \, dz.$$ 

(4)

The Fourier transform can be represented in terms of DFT.
Geometric detail suppression by the Fourier transform: Y G Lee, K Lee

Figure 10 Change of model boundary values due to filtering

by

$$H\left( \frac{l}{LT}, \frac{m}{MT}, \frac{n}{NT} \right) = T_x T_y T_z \sum_{p=0}^{L-1} \sum_{q=0}^{M-1} \sum_{r=0}^{N-1}$$

$$g(pT_x, qT_y, rT_z)e^{-j2\pi \left( \frac{lp}{L} + \frac{mq}{M} + \frac{nr}{N} \right)}$$

(5)

$p = 0, 1, \ldots, L - 1 \quad i = 0, 1, \ldots, L - 1$

$q = 0, 1, \ldots, M - 1 \quad m = 0, 1, \ldots, M - 1$

$r = 0, 1, \ldots, N - 1 \quad n = 0, 1, \ldots, N - 1$

Here function $g(x,y,z)$ is given in Table 1.

In eqn (5) we assume that the function $h(x,y,z)$ has been sampled in the $x,y,z$ dimensions with the sample intervals $T_x, T_y, T_z$, respectively. The resulting sampled function is $g(pT_x, qT_y, rT_z)$. By employing this discrete function we get eqn (6). This is called the DFT (discrete Fourier transform) and an efficient method to compute this is the FFT (fast Fourier transform).

$$G\left( \frac{l}{LT}, \frac{m}{MT}, \frac{n}{NT} \right) = \sum_{p=0}^{L-1} \sum_{q=0}^{M-1} \sum_{r=0}^{N-1}$$

$$g(pT_x, qT_y, rT_z)e^{-j2\pi \left( \frac{lp}{L} + \frac{mq}{M} + \frac{nr}{N} \right)}$$

(6)

$p = 0, 1, \ldots, L - 1 \quad l = 0, 1, \ldots, L - 1$

$q = 0, 1, \ldots, M - 1 \quad m = 0, 1, \ldots, M - 1$

$r = 0, 1, \ldots, N - 1 \quad n = 0, 1, \ldots, N - 1$

The inverse DFT is given in eqn (7).

$$g(pT_x, qT_y, rT_z) = \frac{1}{LMN} \sum_{p=0}^{L-1} \sum_{q=0}^{M-1} \sum_{r=0}^{N-1}$$

$$G\left( \frac{l}{LT}, \frac{m}{MT}, \frac{n}{NT} \right) e^{j2\pi \left( \frac{lp}{L} + \frac{mq}{M} + \frac{nr}{N} \right)}$$

(7)

$p = 0, 1, \ldots, L - 1 \quad l = 0, 1, \ldots, L - 1$

$q = 0, 1, \ldots, M - 1 \quad m = 0, 1, \ldots, M - 1$

$r = 0, 1, \ldots, N - 1 \quad n = 0, 1, \ldots, N - 1$

DFT is a widely used method in the digital signal processing. DFT described above is used to represent a shape by the Fourier basis function, and by low-pass filtering this, detail features are removed.

LPF model and detail suppressed model

Discarding the sinusoidal functions of high frequency reveals a shape that loses the detail features present in the original model. Low-pass filtering has one adverse effect. All sharp corners get rounded. It is customary to leave sharp corners when suppressing detail geometries to consider the stress concentration effect at the sharp corners. To preserve the sharp corners, LPF (low-pass filtered) model is only used as a reference model to evaluate the boundary elements of the original model. The relations between three types of models (original model, LPF model, and the detail-suppressed model) are illustrated in Figure 3.

A boundary element of the original is determined to be negligible or not by calculating its average distance from the LPF model. The average distance is calculated as follows. Sample points are spread on the boundary element to be investigated. From each sample points, the minimum distance to the LPF model is calculated. Then the minimum distances for all the sample points are averaged. For implementation convenience, the LPF model is also sampled by points.

The whole procedure to select the detail boundary element is depicted in Figure 4. In Figure 4(a), given original
model is processed into two folds: the sample points on the simplified model boundary and the sample points on the boundary elements of the original model boundary. Figure 4(b) illustrates the subprocesses for obtaining the sample points on the LPF model boundary. Figure 4(c) illustrates the minimum distance pairs matched by short lines from the sample points on a boundary element of the original model to those of the LPF model boundary. These distances are averaged to assess the detailness of the particular boundary element.

Average distance shows how bad the given boundary element conforms to the LPF model boundary. Thus, we can formulate the following rule; an edge (face in 3D) is part of the overall shape if the following equation is satisfied.

\[
T_0 = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N_i} p_{ij}}{\sum_{i=1}^{M} N_i} \left( \frac{1}{\sum_{i=1}^{O_i} s_{ij} C(T_0)} \right) < \delta,
\]

where \(M\) is the number of edges (faces in 3D) of the geometric model; \(N_i\) is the number of pixels (voxels in 3D) composing an edge (a face in 3D); \(O_i\) is the number of sample points composing an edge (a face in 3D); \(C(T_0)\) is the simplified model boundary pertaining to \(T_0\); \(p_{ij}\) is the value of pixel (voxel in 3D) \(j\) composing an edge (a face in 3D); \(s_{ij}\) is the sample point \(j\) composing an edge (a face in 3D); \(\delta\) is the tolerance of the average distance from \(C(T_0)\).

After each boundary element of the original model is ranked by the average distance to the LPF model, the user can provide a threshold value to classify the boundary elements into the detail elements and those that are not. The threshold value is determined interactively by examining the plot of the average distances. Other methods can be used. For example, the user can give the percentage of boundary elements that the user wishes to delete. Boundary elements that have large average distances are those that have changed a lot by low-pass filtering and they can be concluded to belong to the detail feature. Thus, these boundary elements are deleted and healed. Healing can be challenging for faces, but it is easy for edges. For the detail edges, every edge-loop is traversed in the loop directions and when detail edges are found, the set of connected detail edges are all assembled together and converted to a single edge. From the connected detail edges in the sequence of the loop direction, the starting edge denotes the first edge and the ending edge denotes the last edge. The new edge is formed by connecting the start vertex of the starting detail edge and the end vertex of the ending detail edge.
RESULTS

To DFT a shape, a digitizing is needed. A gripper (Figure 5) was designed in a CAD system and converted to a 2-D digital image of resolution $512 \times 512$. In the figure, the numbers are the edge IDs to be used later. This digital image is shown in Figure 6(a). The black pixels represent the locus of points satisfying eqn (3) and the white pixels represent the surrounding space. The black pixels are assigned the value 1.0, whereas the white pixels are assigned the value $-1.0$. The transition boundary where the value switches from $-1.0$ to 1.0 is assigned the value 0.0. This is the model boundary and $T_0 = 0.0$ in eqn (3).

Using these values as the height, a surface can be plotted [Figure 6(b)]. Figure 7 illustrates the frequency domain plot obtained by Fourier transforming the function in Figure 6(b). It can be observed that the values of the low frequency region (the center region in Figure 7) are large in compared with the high frequency region (the corners in Figure 7).

The plot is only for the real part of the complex frequency domain, but similar result will be obtained for the imaginary part.

We postulated that the low frequency terms form the overall shape and the high frequency terms form the detailing shape. This hypothesis is tested by applying low-pass filtering to Figure 7. Figure 8 is the result after eliminating the frequency greater than the cut off frequency ($\theta$), 0.042017 (justification of this value will be given in the section, setting key parameters). Inverse Fourier transforming Figure 8 to its object space gives Figure 9(a).

Using the height values of the surface in Figure 9(a) as the pixels’ colors, a 2D image can be obtained. In Figure 9(b), the black pixels are where the pixel value is greater than $T_0$ and the white pixels are where it is less than $T_0$. It can be noticed that small pinhole is filled and the teeth are eroded. Figure 9(b) can also be obtained by cutting the surface in Figure 9(a) with a plane on $T_0$. 

Figure 14  Number of sample points on edges

Figure 15  Selected detail edges

Figure 16  Detail removed geometry
LPF (low-pass filtered) model boundary

Low-pass filtering changes the values pertaining to pixels. Our concern is in the model boundary values, which were uniformly assigned to be 0. Figure 10 illustrates these changes graphically. The darker lines show the values are increased and the lighter lines show the values are decreased. Line lengths are amplified by the factor of 50 for visual clarity. Natural question comes out, what is the value of $T_0$ after the filtering? The new model boundary would be the iso-contour of Figure 9(a) pertaining to this value. Thus what we want is to find out a value, which determines the contour such that the shape of this contour does not vary much from the original model boundary. We used the average (0.012129) of the values in the filtered object domain pertaining to the old (original) model boundary elements (pixels) as $T_0$. Figure 11 illustrates the possible contours for several values. In the figure, the thick lines are the original model boundary elements and the thin lines the iso-contours of several values.

The scale accordance between the LPF model boundary and the original model boundary can be calculated as the average distance from the former to the latter. When calculating the average distance between these two boundaries, both boundaries are first converted into two sets of points. Denoting the set of points sampled from the original model boundary as the reference set and the set of points sampled from the LPF model boundary as the testing set, distances from a point from the reference set to the points from the testing set are calculated and the minimum distance is used as the scale accordance. The scale accordance is then calculated as the average of these minimum distances.

\[ \text{Scale accordance} = \frac{1}{n} \sum_{i=1}^{n} d(i) \]

where $n$ is the number of points in the reference set and $d(i)$ is the minimum distance from the $i$th point in the testing set to the points in the reference set.

Figure 17 Triangular mesh generation (a) (b)

Figure 18 LPF model due to various cut off frequency values: (a) $\theta = 0.010504$; (b) $\theta = 0.021008$; (c) $\theta = 0.031513$; (d) $\theta = 0.042017$; (e) $\theta = 0.052521$; (f) $\theta = 0.063025$

Figure 19 Korean peninsula and the coastline model (a) (b)
value of the calculated distances is saved. This process is applied to the remaining points in the reference set. At the end, the average distance is calculated as the average of the saved distances.

If this average distance between a selected contour pertaining to a particular value and the original model boundary is smaller than any other such contours pertaining to other values, the value that determines the contour exhibiting the minimum distance from the model boundary is what we are looking for. In other words, LPF model is now in scale accordance to the original model.

This value can be obtained through iterations. However, we chose not to do computationally expensive iterations. A reasonably approximate value can be obtained by averaging the values in the filtered object domain at the old (original) model boundary elements (pixels) and using this value as the new $T_0$. This is the best answer that can be obtained without doing a rigorous test. The new $T_0$ was found to be 0.012129 for our 2D example. Iso-contour pertaining to $T_0$ is not clearly marked in the Figure 11 as it is hidden by the original model boundary elements.

A nice geometric interpretation can be made about $T_0$ by a one-dimensional example illustrated in Figure 12. In the old value distribution, the base value is −1.0 and the ridge value is 1.0. For the transition region between the two values, we assume a continuous change having the slope of 45° as illustrated in Figure 12. Note that we need a continuous function to apply Fourier transform. Figure 12 shows the temperature distribution (remember we previously assumed a life form temperature) corresponding to a line segment, that is the region having the value greater than the old $T_0$ corresponds to the line segment. After the low-pass filtering, the temperature distribution is changed to the new value distribution. If the old $T_0$ is used to get the modified life form or the new geometric model, the resulting model will be scaled larger. To get the shape as close to the old model as possible, the new $T_0$ must compensate this scale effect. This can be accomplished by adding the average of value changes at each boundary points to the old $T_0$ to get the new $T_0$.

Selection of detail edges

Consider the average distance from the edges of the original model to the LPF model boundary. Figure 13 plots the average distances for each edge from the LPF model boundary pertaining to $T_0$. The horizontal axis stands for the edge IDs and the vertical axis is the average distance value.
The unit of the distance is the distance between two consecutive pixels in the horizontal or vertical directions. The edge IDs were given in Figure 5. The IDs are numbered in the clockwise direction for each loop (there are one peripheral loop and one hole loop). Figure 14 shows the number of sample points on the edges. The number of sample points is proportional to the length of the corresponding edge. We can conclude the following rule by comparing Figures 13 and 14. Short edge length does not mean the corresponding edge is detail.

By using the values (definition and justification of these values will be given in the next section) in Table 2, the detail edges are classified and deleted. $T_0$ (Discriminating value between void and volume) was set to be 0.0 and later calculated to be 0.012129. The edges determined to be the detail edges are indicated in Figure 15 by thin lines. The small pinhole and the teeth are classified as the detail edges. The detail-suppressed gripper is shown in Figure 16. Figure 17(a) illustrates the mesh generated on the detail-removed gripper. For the comparison, Figure 17(b) illustrates the mesh generated on the original model. Mesh element number is reduced from 347 to 230. Which is a 33.7% reduction. The CPU times for FFT/IFT, detail edge detection, and boundary reconstruction altogether for a 512*512 image took 53 s on an engineering workstation with MIPS R4400 running at the clock speed of 150 MHZ.

Setting key parameters

There are two numbers used in the previous 2D example that need explanations. To be brief, the two numbers, $\theta$ (cut off frequency) and $\delta$ (tolerance of the average absolute deviation from $T_0$) are given by the user, but the choices of their values are not arbitrary.

As for $\theta$ (cut off frequency), the user is provided with several outputs for different cut off values and allowed to choose the proper value. Example illustrations are given in Figure 18. Readers can be persuaded that the natural choice would be Figure 18(d), which is what we chose. The user would use larger cut off value to be more conservative, i.e. retain more edges, or use smaller one to delete more detail edges.

Now we discuss about $\delta$ (tolerance of the average distance from the LPF boundary). Given the plot in Figure 13, it can be seen that the detail edges and the remaining edges are quite sparsely distributed. The gray edges that bother us are the transition edges, which are in between the detail edges and the non-detail edges. It would be reasonable to select somewhere around 2.0. We chose 1.5 to push the gray edges to the detail side, it would be also possible to use a larger value and pull the gray edges to the non-detail side.

Complex example

To demonstrate the robustness of the proposed method in case of complex geometric models, we chose the satellite picture [Figure 19(a)] of the Korean peninsula and detected the coastline [Figure 19(b)] using one of the edge detecting algorithms. The coastline information can be used to simulate the movement of typhoons or to forecast the weather. Automatic edge detection often produces too many edges. Thus, detail geometric features in a complex model like this cannot be effectively removed by human. The purpose of applying our method is to reduce the number of edges as well as removing some small islands. These are insignificant features for the simulation.

Same procedures are performed. The original coastline model is digitized and low-pass filtered to obtain the isocontour illustrated in Figure 20, the cut off frequency selected was $\theta = 6.683$. By computing the average distances of the original coastline edges to the LPF model, plot of Figure 21 is obtained.

Using the threshold distance value of 0.02, the detail edges are identified and removed. The edge numbers were reduced from 600 to 473 (21.6% reduction). Triangular meshes are generated on the original model [Figure 22(a)] and the detail suppressed model [Figure 22(b)]. Mesh elements were reduced from 1706 to 1200 (23.8% reduction).

DISCUSSION

Two topics in relation to the method of signal processing will be discussed here. They are the ringing effects and the
use of a modern signal processing technique called wavelets. We will also discuss about the robustness of the proposed algorithm.

**Monotonic smooth filter**

We used a box filter in the frequency domain for the low-pass filtering. Box filters are known to cause rippling (ringing) effects. It is common to use other filters instead of the box filter to reduce these effects. Figure 23 illustrates one of these filters, a monotonic smooth filter, represented by the following equation.

\[
h(u, \alpha) = \frac{1 + \cos \left( \frac{\pi u}{\alpha} \right)}{2} \quad \text{when } |u| < \alpha \quad (9)
\]

\[
h(u, \alpha) = 0 \quad \text{when } |u| \geq \alpha.
\]

\(\alpha\) is used to control the cut-off frequency \(\theta\), where

\[
\theta = \cos^{-1} \left( 1 - \frac{1}{\alpha^2} \right)
\]

Monotonic smooth filter biases the magnitude of the frequency components by their proximity to the cutoff frequency. In other words, frequency components close to...
the cut-off frequency are much more decreased than those that are far from it. Figure 24 illustrates the simplified (low-pass filtered) models using this filter for various cut-off frequencies and Figure 25 illustrates the filtered object domain view. It can be noticed that the rippling effects shown in Figure 9(a) does not exist in Figure 25. However, there are little improvements in the iso-contours of these object domain surfaces, which are what really matter. In summary, rippling effects present in the surfaces do not persist in the iso-contours and it is good enough to use the box filter.

Wavelets

Fourier basis functions have non-zero value in the entire domain. There is a family of basis functions that have narrow range of non-zero values (small support). They are called wavelets. The basis functions of the Fourier transform are sinusoidal functions and these functions span in the entire object domain. In other words, they have a wide support. On the contrary, wavelets have a small support. Wavelets are known to work better than the Fourier basis functions in some applications for its small support. We tested the 2D gripper with Daubechies 4 (Figure 26), 12 (Figure 27), 20 (Figure 28) wavelets for various remaining percentages of the detail coefficients. Corresponding results are illustrated in Figures 29, 30, and 31. Specified percentage of detail coefficients with large values is retained while the small detail coefficients were set to zero.

The results show that narrow support characteristic of wavelets is not suitable for removing detail features. This can be explained that every feature, no matter how small, should influence the overall shape of the model. Thus the basis function describing a small local feature should influence the entire domain rather than local to its vicinity. Figure 32 illustrates the filtered object domain using wavelets. Daubechies 12 with 0.99832% of coefficients was used for the view.

Robustness

We think that the detail boundary element selection scheme is quite robust. If something would go wrong, it would be in reconstructing the boundary elements after deleting the detail edges. A new edge formed at the place of the wound can intersect the remaining non-detail edges. This can happen between a hole loop and the peripheral loop as depicted in Figure 33(a). In the figure, initial upper state is changed to the lower state by removing the box shaped protrusion. Consequently, the new peripheral loop intersects the inner loop. To overcome this, the new edge must check that it does not intersect with the remaining edges and if it does, the deleted edges must be revived. It is hard to imagine

Figure 24  Filtered object domain using monotonic smooth filter

Figure 25  Daubechies 4 wavelets
the new edge intersecting the loop that it is part of. Figure 33(b) illustrates this situation. However, for robustness, this situation should also be considered.

CONCLUDING REMARKS

By suppressing the small features prior to the mesh element generation, computing time can be reduced. Unnecessary elements are intelligently removed for greater efficiency in utilizing the computer resources. This article also has another potential application. Commonly, features are studied in two different approaches. One is the activity for designing with features and the other is for recognizing features by comparing adjacent topological entities. Little has been known about looking at features by size. This article proposes a metric to characterize a feature through its size.

Question may arise, would it be necessary to automate the selection of detail features? The answer we give is as follows. Our metric system provides an unambiguous method to rank features according to its size. Yes, an experienced user may select detailing features, but even so, a metric...
system can assist the user, not to mention the difficult situation when it is ambiguous as to which one to delete prior to the others.

This article does not provide a fully automatic algorithm to suppress detail boundary elements with estimates of the errors resulting from these simplifications. However, we think the algorithm proposed here can be used in conjunction with an analysis engineer in the following manner. The system proposes detail features to the engineer. The engineer manually overrules some of the selections that (he thinks) are important to the analysis. The engineer finally adds more features that are not selected by the system but are thought to be unimportant to the analysis.

The difficult part in extending the proposed algorithm to 3D is in reconstructing the boundary after the detail faces are classified and deleted. This is a difficult operation since there are many possible methods to fill a wound (where the faces are deleted). We are developing a triangulation technique to heal these wounds. There is also a concern in the performance issue when the proposed algorithm is applied to a 3D volume. These problems are being further investigated.

![Figure 29](image1.png)

**Figure 29** Filtered model for various remaining percentages of detail coefficients for Daubechies 4 wavelets: (a) 7.815552%; (b) 1.013184%; (c) 0.099945%; (d) 0.010300%

![Figure 30](image2.png)

**Figure 30** Filtered model for various remaining percentages of detail coefficients for Daubechies 12 wavelets: (a) 8.902740%; (b) 0.999832%; (c) 0.099945%; (d) 0.010300%
ACKNOWLEDGEMENTS

This work is supported by Korean Ministry of Education through Research Fund ME96-E-17.

REFERENCES


Kunwoo Lee is a professor in the Department of Mechanical Design and Production Engineering at Seoul National University in Korea. He received his BS (1978) from Seoul National University, and MS (1981) and Ph.D. (1984) in mechanical engineering from MIT. He was previously an assistant professor in the Department of Mechanical and Industrial Engineering, University of Illinois at Urbana–Champaign, and a visiting associate professor in the Department of Mechanical Engineering at MIT. Currently, he is on the editorial board of the journal Computer-Aided Design. He is also a vice president of the Society of CAD/CAM Engineers in Korea. His research interests include assembly modeling, non-manifold modeling, finite element mesh generation, topology optimization, and rapid prototyping.

Yong-Gu Lee is an advisory engineer at Samsung SDS. He received a BS (1992), MS (1994), and Ph.D. (1997) in mechanical design and production engineering from Seoul National University, Korea. He was a research assistant of Institute of Advanced Machinery and Design at Seoul National University from 1994 to 1997 and participated in several government and industry sponsored projects. Those are ‘development of CAD kernel system’, ‘development of an integrated feature based modeling system for mechanical part design’, and ‘feature based modeling system for design and analysis of tank’. His research interests include computational geometry, finite element mesh generation, and product data exchange. His current research activity focuses on geometry simplification, such as dimensional reduction (medial surface) and detail suppression.